EXHIBIT A

A Multicast Routing Protocol for Ad-Hoc Networks

J.J.Garcia-Luna-Aceves and Ewerton L. Madruga
{jj, madruga}@cse.ucsc.edu
Computer Engineering Department
Baskin School of Engineering
University of California
Santa Cruz, CA 95064

Abstract—The Core-Assisted Mesh Protocol (CAMP) is introduced for multicast routing in ad-hoc networks. CAMP generalizes the notion of core-based trees introduced for internet multicasting into multicast meshes that have much richer connectivity than trees. A shared multicast mesh is defined for each multicast group; the main goal of using such meshes is to maintain the connectivity of multicast groups even while network routers move frequently. CAMP consists of the maintenance of multicast meshes and loop-free packet forwarding over such meshes. Within the multicast mesh of a group, packets from any source in the group are forwarded along the reverse shortest path to the source, just as in traditional multicast protocols based on source-based trees. CAMP guarantees that, within a finite time, every receiver of a multicast group has a reverse shortest path to each source of the multicast group. Multicast packets for a group are forwarded along the shortest paths from sources to receivers defined within the group's mesh. CAMP uses cores only to limit the traffic needed for a router to join a multicast group; the failure of cores does not stop packet forwarding or the process of maintaining the multicast meshes.

Keywords—Multicast routing protocol, ad-hoc networks, multicast mesh, mobility.

I. Introduction

With few exceptions, the methods used today for supporting many-to-many communication (multicasting) efficiently in computer networks involve routing trees. The basic approach consists of establishing a routing tree for a group of routing nodes (routers). Once a routing tree is established for a group of routers, a packet or message sent to all the routers in the tree traverses each router and link in the tree only once. Multicast routing trees (multicast trees for short) are being used extensively for multicast routing in computer networks and internets [1], [7], [8], and have also been proposed for wireless multi-hop networks [2], [4], [12].

Because a multicast tree provides a single path between any two routers in the tree, the minimum number of copies per packet are used to disseminate packets to all the receivers of a multicast group. For a tree of N routers, only $N-1$ links are used to transmit the same information to all the routers in the multicast tree in a network with point-to-point links; in the case of wireless networks with broadcast links using a single channel, each member of a multicast tree needs to transmit a packet only once. Using routing trees is of course far more efficient than the brute-force approach of sending the same information from the source individually to each of the other $N-1$ times. An additional benefit of using trees for multicast routing is that the routing decisions at each router in the multicast tree becomes very simple: a router in a multicast tree that receives a multicast packet for the group over an in-tree interface forwards the packet over the rest of its in-tree interfaces.

However, multicast trees achieve the efficiency and simplicity described above by forcing a single path between any pair of routers. Accordingly, if multiple sources must transmit information to the same set of destinations, using routing trees requires that either a shared multicast tree be used for all sources, or that a separate multicast tree be established for each source. Using a shared multicast tree has the disadvantage that packets are distributed to the multicast group along paths that can be much longer than the shortest paths from sources to receivers. Using a separate multicast tree for each source of each multicast group forces the routers that participate in multiple multicast groups to maintain an entry for each source in each multicast group, which does not scale as the number of groups and sources per group increases. In addition, because trees provide minimal connectivity among the members of a multicast group, the failure of any link in the tree partitions the group and requires the routers involved to reconfigure the tree.

An ad-hoc network is a packet-switching network based on wireless links for router interconnection. The topology of an ad-hoc network can be very dynamic due to the mobility of routers and the characteristics of the radio channels. Although tree-based multicast routing is very attractive for wired networks and the Internet because of its simplicity, we argue that it is not as applicable to ad-hoc networks with dynamic topologies. Maintaining a routing tree for the purposes of multicasting packets when the underlying topology changes frequently can incur substantial control traffic. Furthermore, during periods of routing-table instability, routers may be forced to stop forwarding packets while they wait for the multicast routing tree to be reconstructed. This paper focuses on multicast communication in ad-hoc networks and presents the first generalization of routing trees into graphs that have more connectivity than trees and yet prevent long-term or permanent routing loops from occurring. We call these routing graphs *multicast meshes*. The key contributions of this paper consist of proving that it is possible to establish and maintain routing structures for multi-point communication in an ad-hoc network that are far more resilient than trees and can make efficient use of communication resources, and presenting the first multicast routing protocol that uses routing structures different than trees, without the need to first flood an entire network or internet with either data packets (like DVMRP or PIM-DM do), or control packets (like the Forwarding Group Multicast Protocol (FGMP) does [3]).

Section II introduces the main design principles of the Core-Assisted Mesh Protocol (CAMP), which builds and maintains shared multicast meshes, and routes packets from any group source over the shortest from source to receivers defined in the group's mesh. Section III to VIII describe in more detail the creation and maintenance of multicast meshes and the techniques used for packet forwarding in CAMP. Section IX describes the results of simulation experiments used to study CAMP's performance compared to tree-based multicasting in a dynamic topology. Section X provides our concluding remarks.

II. Overview of CAMP

The Core-Assisted Mesh Protocol (CAMP) is designed to support multicast routing in very dynamic ad-hoc networks with broadcast links. It adopts the same basic architecture used in IP multicast [6]. A mapping service is assumed to exist that provides routers with the addresses of groups identified by their names. In the Internet, this service would be provided by the Domain Name System (DNS), for example. Hosts wishing to join a multicast group must first query the mapping service to obtain a group address and then interact with their local routers (which we call routers here) through IGMP [6] or an equivalent host-to-router protocol to request membership in a multicast group. In addition to the naming service used by hosts to obtain multicast group

This work was supported by the Defense Advanced Research Projects Agency (DARPA) under Grant F30602-97-2-0338 and by CNPq-Brazil.

addresses, CAMP assumes the availability of routing information from a unicast routing protocol. The only requirement imposed on the routing protocol used is that it must provide correct distances to known destinations within a finite time.

CAMP differs from most prior multicast routing protocols in that it builds and maintains a *multicast mesh* for information distribution within each multicast group. A multicast mesh is a subset of the network topology that provides at least one path from each source to each receiver in the multicast group. CAMP ensures that the shortest paths from receivers to sources (called reverse shortest paths) are part of a group's mesh. Packets are forwarded through the mesh along the paths that first reach the routers from the sources, i.e., the shortest paths from sources to receivers that can be defined within the mesh. CAMP does not predefine such paths along the mesh. A router keeps a cache of the identifiers of those packets it has forwarded recently, and forwards a multicast packet received from a neighbor if the packet identifier is not in its cache and the router has been told by at least one neighbor that the router is the successor in a reverse shortest path to any group source.

Because a member router of a multicast mesh has redundant paths to any other router in the same mesh, topology changes are less likely to disrupt the flow of multicast data and to require the reconstruction of the routing structures that support packet forwarding. Figure 1 illustrates the differences between a multicast mesh and the corresponding shared multicast tree; routers that are members of the multicast group are dark. The multicast mesh and tree shown in the figure include routers that have host receivers, hosts that are senders and receivers, and routers that act only as relays. Router *g* is the last receiver to join the multicast group, and does so in the multicast mesh through either router *f* or *h*; consequently, router *c* does not become a member of the mesh.

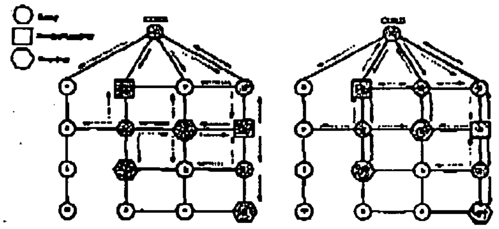

Fig. 1. Traffic flow from router *h* in a multicast mesh (left) and in the equivalent multicast shared tree (right).

Figure 1 illustrates how data packets are forwarded from router *h* to the rest of the group in CAMP and in a shared-tree multicast protocol. In the figure, solid arrows indicate the flow of traffic along the reverse shortest path in CAMP, and along a shared-tree in the multicast protocol; dashed arrows indicate overhead traffic due to the broadcast characteristics of the communication channel used to connect them. Note that CAMP delivers data along shorter paths than a shared-tree multicast protocol. The length of paths incurred in multicasting over ad-hoc networks is very important, because longer paths require more routers forwarding packets. It is also interesting to note that, in this example, the number of routers receiving the packets sent by *h* at least once is the same using the multicast mesh or the shared-tree; this shows that using multicast trees instead of meshes does not necessarily reduce traffic overhead in ad-hoc networks with broadcast links.

CAMP extends the basic receiver-initiated approach introduced in the core-based tree (CBT) protocol [1] for the creation of multicast trees to enable the creation of multicast meshes. Cores are used to limit the control traffic needed for receivers to join multicast groups. In contrast to CBT, one or multiple cores can be defined for each mesh, cores need not be part of the mesh of their group, and routers can join a group even if all associated cores become unreachable. A router sends a join request towards a core if none of its neighbors are members of the group; otherwise it simply announces its membership using either reliable or persistent updates. If cores are not reachable from a router that needs to join a group, the router broadcasts its Join request using an *expanded ring search* (ERS) that eventually reaches some group member. When one or multiple responses are sent back to the router, it chooses any path to the mesh. Routers can leave the group simply by advertising the change in group membership to their neighbors.

The Forwarding Group Multicast Protocol (FGMP) [3] also builds a variation of meshes. However, to establish group meshes, FGMP requires for control packets to be flooded in an ad-hoc network. This approach is acceptable only in small networks. In contrast, the use of cores in CAMP eliminates the need for flooding, unless all cores are unreachable from a connected component.

CAMP uses a scheme based on the transmission of *heartbeat* messages to ensure that the mesh contains all the reverse shortest paths. Each mesh member temporarily keeps track of traffic sources whose packets come through members other than their respective reverse shortest paths to the sources. Whenever such situation arises, a heartbeat is sent to the successor in the reverse shortest path to the source given by the unicast routing table. That heartbeat message triggers a *push join* message when the successor is not a mesh member. The *push join* forces that specific successor and all routers in the path to the traffic source to join the mesh. Mesh components merge together by means of similar push joins sent towards cores.

The mappings of multicast addresses to (one or more) core addresses are disseminated from each core out to the network as part of group membership reports.

III. Routing Information Maintained in CAMP

Each router maintains a routing table (RT) built with the unicast routing protocol used in conjunction with CAMP, a Core-to-Group Address Mapping (CAM) table, an anchor table (AT), a multicast routing table (MRT), and a packet-forwarding cache.

At router *i*, the RT made available to CAMP specifies, for each destination *j*, the successor (s^i_j) and the distance to the destination (D^i_j).

Router *i*'s CAM consists of a vector of core-to-group address mappings. Each entry of the CAM specifies a group address and the addresses of the cores that can be contacted for that group.

Router *i*'s AT has an entry for each of the multicast groups in which router *i* is a member. For each multicast group *g*, an entry in the AT specifies those neighbors that router *i* uses as its *anchors* for the group, and whether the router has any local host that is a source or receiver of the group. An anchor for router *i* in group *g* is a neighbor router that is a successor (next hop) in the reverse shortest path to *at least one source* in the group *g*. Each anchor entry consists of the address of the neighbor and an age field. In the example shown in Figure 1, router *f* uses router *g* as an anchor for the group. Note that a router does not maintain anchor information for each source in a group.

The MRT at router *i* lists, for each multicast group address *g* known to router *i* and for each neighbor *n* that has advertised being a member of the group, an *anchor flag* stating whether *i* is one of *n*'s anchors in *g* and a *membership type flag* stating whether *n* is a simplex or duplex member of the group. A router joins a group in simplex mode if it intends to only send traffic received from specific attached hosts or neighbor routers to the rest of the group, and it does not intend to forward packets from the group. Duplex membership implies that the router forwards any multicast packet for the group.

The packet-forwarding cache maintains the identifier of packets recently forwarded by the router. Basically the information kept in this data structure comes straight from the IP packet header, which are source address, destination address (group address), packet identification and fragment offset. The address of the neighbor that relayed that packet is also stored. The main role of the packet forwarding cache is to avoid packet replication by keeping track of packets already received by the router.

When router *i*'s MRT or AT is updated, it sends a multicast routing update (MRU) to all its neighbors reporting changes in its group membership and anchors per group. An MRU contains one or more entries, and each entry specifies:

- A group address.
- An operation code specifying a quit notification, simplex membership notification, or a duplex membership notification.
- In membership notifications, the list of anchors needed by router *i* for the group.

In an ad-hoc network, changes in multicast-group memberships should be disseminated together with routing-table updates, and routers hear the reports from their neighbors and remember which neighbors belong to which group. To save bandwidth, routers should exchange multicast routing information together with their unicast routing-table updates. Hence, a routing-table update would consist of a unicast part and a multicast part. However, we describe CAMP independently of the unicast routing protocol with which it is used.

The main objective of communicating anchor information among routers is to prevent routers that are required by their neighbors to forward multicast packets from leaving groups prematurely.

A router may update its MRT or AT after topology changes and messages received from its neighbors. The messages that may trigger an MRU are MRUs from neighbors that change group memberships, ACKs that change both membership and anchor information.

Detecting the failure or addition of a link to a neighbor is part of the routing protocol used in conjunction with CAMP. For CAMP to work correctly, it is necessary for the associated routing protocol to work correctly in the presence of router failures and network partitions. This implies that CAMP cannot be used in conjunction with a routing protocol based on the distributed Bellman-Ford algorithm such as the routing protocol of the DARPA packet radio network [11]. However, there are several recent examples of routing protocols that can be used in conjunction with CAMP [5], [13], [14], [16].

IV. Basic Joining and Quitting Mechanisms

We assume in this section that each router can reach at least one core of the multicast group that the router needs to join, and that the multicast mesh is connected.

CAMP uses a receiver-initiated method for routers to join multicast groups that differs from CBT's join mechanism in a number of ways. A host first determines the address of the group it needs to join as a receiver. The host then uses that address to ask its attached router to join the multicast group. Upon receiving a host request to join a group, the router then determines whether to *announce* its membership in the group or to *request* being added to the group, and uses its CAM to select the core towards which the join request may be sent. In CBT, joining a group always implies a request to join, and a router selects the relay of a join request as the neighbor router along the shortest path to the group core.

If a router joining a group has multiple neighbors that are *duplex* members of the multicast group, then the router simply changes its MRT and directly announces to its neighbors that it is a new member of the multicast group using an MRU. The announcements states whether the router is a simplex or duplex member. If MRUs are sent reliably (depending on the unicast routing protocol) the neighbor nodes acknowledge the join announcement. If MRUs are sent unreliably, the join announcement is sent periodically, so that neighbors learn about the join over time.

If a router joining a group has no neighbors that are members of the multicast group, then it selects its successor to the nearest core as the relay for the join request. After the router selects a relay, it sends a *join request* to all its neighbors. A join request specifies the intended relay, the address of the multicast group that the sending router needs to join, and whether the router wants to join in simplex or duplex mode.

After sending a join request, a router then waits for the first acknowledgment to its request, and retransmits the request after a request-timeout. The router persists sending join requests for a number of times (e.g., four) as long as the unicast routing table indicates that there are physical paths towards *any* of the group cores and none of its neighbors are group members. Each retransmission of a request is addressed to an intended relay chosen as described above. This is similar to the basic mechanism used in CBT; however, because data packets flow along different paths over the multicast mesh depending on the source, there is no need to ensure a single loopless path to the chosen core. This means that the use of selected relays towards any core is simply used to *limit* the search from the routers towards the multicast mesh, but being able to reach a core is not necessary to join a group.

Any router that is a regular member of a multicast group and receives a join request for the group is free to transmit a *join acknowledgment* (ACK) to the sending router. An ACK specifies the sender of the join request and the multicast group being joined. To reduce channel traffic, the router specified as the relay of a join request can be allowed to reply first by means of a timeout mechanism after a join request is received.

When the origin or a relay of a join request receives the first ACK to its request or the first ACK to a join request for the same multicast group, the router becomes part of the multicast group. In the case of a relay, the router sends an ACK to the previous relay or origin of the join request, even if that neighbor has already sent an update stating that it is a member of the multicast group.

Receivers use a slightly different procedure to leave a multicast group in CAMP than in CBT. A router leaves a multicast group when it has no hosts that are members of the group *and* it has no neighbors for whom it is an *anchor*.

A router leaving a multicast group issues a *quit notification* to its neighbors, who in turn can update their MRTs. No acknowledgments are requested for quit notifications, because in contrast to multicast routing trees, multicast meshes do not dictate the paths taken by multicast packets. Quit notifications are sent as part of multicast routing updates.

In an ad-hoc network, it is likely that the routers serving as access points to the rest of the Internet would serve as cores, because they are static and they must be known by the rest of the ad-hoc network for other purposes. An interesting feature of CAMP is that cores are allowed to leave multicast meshes, as long as there are no routers using them as anchors, i.e., as as long as they are needed to provide efficient paths for the dissemination of packets in the multicast meshes of the groups.

In the example shown in Figure 1, the core router could leave if *b* and *d* did not use it as an anchor. This would be the case if *c* joined the multicast mesh. An approach to favor non-core routers as anchors consists of using a routing protocol that provides multiple paths to the same destination, and making CAMP use non-core successors whenever possible.

V. Simplex Joins

If non-member routers were allowed to send packets to a multicast mesh, the only way to reach the mesh without flooding would be through one of its cores. Accordingly, cores could become hot spots if multiple non-member sources existed, and the paths followed by the packets sent by those sources could be very inefficient due to router mobility in an ad-hoc network. Unlike other protocols that allow non-member routers to send packets to a multicast tree for dissemination within the tree, CAMP requires that the router attached to any source of packets for the group join the multicast mesh. To avoid the dissemination of multicast packets to routers that join a group only to allow a source-only host to send packets to the group, CAMP allows routers to belong in a multicast mesh in simplex mode, rather than as regular members. This characteristic of mesh members is used during packet forwarding to avoid the dissemination of data to sender-only routers.

In order to adapt also to bursty traffic, the router connected to the source host does not discard data packets until it receives an acknowledgment for his join request in simplex mode. The router encapsulates data packets into multiple copies of its simplex-mode join requests. Those encapsulated packets are sent towards the core as any other join requests. To minimize the chances of making the core a hot-spot, the first router in the path from the source of data traffic to the core that is already a member starts forwarding the data packets itself. In the worst case, all routers in the path to the core are not mesh members, and the core has to be involved in the packet forwarding while the router closest to the core gets its acknowledgment to the join request.

Figure 2 illustrates the benefits of having members that forward data in one direction only. In CAMP, routers *m*, *i*, and *e* join the group in simplex mode, and forward traffic from host A to the rest of the mesh. In contrast, in a shared-tree protocol, routers *m*, *i*, *e*, and *a* would forward packets from host A to router *z*, the core, and not join the group. It is clear from this example that the approach used in CAMP leads to shorter delays in the distribution of packets from non-member hosts and less congestion at the core routers; furthermore, because of the simplex memberships in the mesh, traffic from other sources does not flow to non-member sources. Dotted arrows are used again to indicate overhead traffic due to the broadcast characteristics of the communication channel, and solid arrows indicate the traversal of packets accepted at each relay and end point. Simplex routers are shown in bold circles. It is important to note that cores in CAMP do not need to be part of of a multicast mesh. In Figure 2, the mesh could exist even if *z* were not part of it because traffic flows along reverse shortest paths and source-only nodes are part of the mesh through unidirectional paths. In contrast, in CBT for example, the core is contacted by traffic from source-only nodes and the shared-tree breaks when the core fails. That is, considering the shared-tree example of Figure 2, node *z* must be a member of the shared-tree if traffic from non-members is to be propagated to the multicast group.

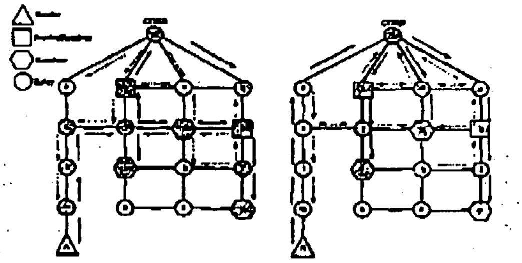

Fig. 2. Traffic flow from non-member sources in CAMP (left) and a shared-tree protocol (right).

VI. Heartbeats, Push Joins, and Anchors

CAMP ensures that all the reverse shortest paths between sources and receivers are part of a group's mesh by means of *heartbeat* and *push join* (PJ) messages.

Periodically, every single entry in the packet forwarding cache is verified. The router looks up its RT to check whether the neighbor that relayed the packet is the reverse path to the source for every cache entry. A heartbeat or a PJ is sent towards every source stored in the cache that had the number of packets coming from the reverse path under a threshold.

A router receiving a heartbeat for a given multicast group and source retransmits the heartbeat if its successor towards the origin of the heartbeat (determined with the unicast routing protocol) is already a mesh member. When a member router receives a heartbeat and detects that its successor is not part of the multicast mesh, it sends a *push join* (PJ) to that neighbor router and waits for an ACK from that router.

Alternatively, if the reverse-path successor for a source of an accepted multicast packet is not a mesh member. the router sends a PJ to that neighbor router and waits for an ACK from that router.

A router retransmits a PJ after a request timeout, and persists sending the PJ, until the unicast routing table indicates that there is no path to the origin of the heartbeat. A PJ specifies:

- The multicast group being joined.
- The intended relay of the PJ.
- The end point of the PJ (i.e., the address of a router with an attached source).

If an ACK to a PJ is needed from a neighbor and the link to that neighbor fails, the router sends a new PJ to a different neighbor using the updated information in its unicast routing table.

A router that receives a PJ sends an ACK if: (a) it is the intended relay, (b) it is already a member of the group specified in the PJ, and (c) it has a path to the end point of the PJ. A router sending an ACK to a neighbor's PJ understands that it is a group anchor for that neighbor.

A router receiving a PJ forwards it to the next relay if: (a) it is the specified intended relay and (b) it has a path to the end point of the PJ. The relay specified in the forwarded PJ is the router's successor to the end point of the PJ. A router discards a PJ for which it is not the intended relay or for which it is the intended relay but has no path to the end point of the PJ.

Heartbeats are sent as long as the reverse shortest path is quiet and anchor information is aged accordingly to account for the fact that the reverse shortest paths used for data distribution change over time, because sources may leave groups and routers may move frequently. When a router stops seeing traffic, it obviously stops forwarding data packets. This causes some of the anchors stored at member routers to age out. This in turn reduces the number of copies of the same multicast packets from other sources received by some routers, and may also allow some routers to leave the group.

After topology changes, the reverse shortest paths from sources to members of the group change, which obsoletes some of the anchors stored at routers. However, packet forwarding in CAMP depends only indirectly on the reverse-path information obtained from the routers' unicast routing tables. Anchor information is used only to prevent routers to leave a multicast mesh when they are in the paths between sources and receivers in the mesh, and packets flow along the shortest paths within the mesh. Accordingly, it is acceptable for a router to attempt to add anchors as quickly as possible (i.e., as soon as it detects a heartbeat from a router for which the successor is not in the multicast mesh), and to wait for anchor information to age out for deletion.

Router *i* can add a neighbor *p* as an anchor for group *g* in two ways after *i* receives a *heartbeat* or a *push join* associated to some source *S* in *g*:

- When *p* forwards an acknowledgment to the *push join* and is also the successor for *i* in the reverse path to source *S*. When router *i* forwards the acknowledgment, it also sends a multicast update if *p* became an anchor.
- When router *i* gets data packets from router *p*, which is also a successor for *i* in reverse path to source *S*.

Anchors are aged while they are stored in the AT and MRT, and are erased when they reach a 0 age. A router can leave a multicast mesh when its MRT shows that no neighbor uses it as an anchor and it has no attached hosts that are senders or receivers of the group.

VII. Handling Topology Changes

A. *Link Failures*

Link failures are not very critical in CAMP. When a link fails breaking the reverse shortest path to a source, the router affected by the break may not have to do anything, because the new reverse shortest path may very well be part of the mesh already; furthermore, packets keep flowing along the mesh through the remaining paths to all receivers. In contrast. if any branch of a multicast tree fails, the tree must reconnect all components of the tree for packet forwarding to continue to all receivers.

Link failures have a smaller effect in CAMP than in tree-based multicast protocols, because a router joins a group with the first ACK it receives from any neighbor, and because a router persists in joining while it has neighbors that are members of the mesh or its unicast routing table provides a path to a core. Furthermore, core failures do not interrupt packet forwarding in the mesh or the ability of new members to join a group, because ERS can be used to reach a multicast mesh when cores are not reachable, and cores need not be part of the mesh. In contrast, in tree-based multicast routing protocols based on receiver-initiated joining (e.g., CBT and PIM-SM), failure of the core or rendezvous point of the group breaks the multicast tree, and prevents new members from joining, until a new one is elected and made known to all routers.

B. Node Failures

CAMP reduces control traffic associated with the establishment and maintenance of multicast meshes by using multiple cores per group that routers can use as *landmarks* to orient their join requests. Therefore, a router can try to join a mesh orienting its unicast join requests to any of such *landmarks* and can redirect its join requests when topology changes. If none of the cores of a group are reachable given the unicast routing information currently available when a router needs to send a join request, this router uses an Expanded Ring Search (ERS) to reach the mesh. This router first sends a mesh search message specifying itself as the requester. Any router receiving such a message forwards it appending its ID to the path of the message, if the ERS can proceed and the router is not a member of the mesh. A router that receives the mesh search message and is a mesh member replies with an acknowledgment. When the mesh search requester gets the first acknowledgment to its message, it sends a *join* request along the path it obtained with the acknowledgment. The router retransmits its search message after a time out if it does not receive an ACK.

Hence, CAMP has no single point of failure and can use as many cores as desired for a given mesh. In contrast, CBT and PIM-SM require a single core to be used in order to provide a multicast tree at all (i.e., avoid to detect loops in the multicast tree and detect partitions). ERS could be used when the single core fails, of course, but that still leaves CAMP as a more efficient approach, because ERSs are used less often by providing multiple cores (no single point of failure). A proposal to accommodate multiple cores and still provide multicast trees has been made recently [15], but the mechanisms in such a proposal appear much too complex for a dynamic network and no similar solutions have been proposed for ad-hoc networks [4], [12].

C. Keeping Meshes Connected

A multicast mesh may be partitioned due to the mobility of routers or the partition of the network itself. In such a case, CAMP has the ability to continue the operation of all mesh components, because routers do not rely on a single core to join the mesh. In any tree-based protocol based on receiver-initiated joining, the tree component including the core or rendezvous point can continue to operate, but the other must terminate the multicast group (or make use of ERS, for example, for every join request) until a path to the core is re-established.

In addition, CAMP is able to merge mesh components as long as there is physical connectivity between mesh components. The mechanism used to accomplish this is simple and is based on requiring each router to keep a record of all the cores in a multicast group, even when the cores are not reachable.

When a router looses connectivity with all the cores of a multicast group, it sets a flag to remember to contact any such core when the unicast routing tablo indicates that at least one core for the group is reachable. When a router detects that connectivity with at least one core of the multicast group is re-established, it determines if its successor in the reverse shortest path to the core is in the mesh, and sends a PJ towards the core if the successor is not in the mesh; this use of PJs reconnects a multicast mesh.

To ensure that two or more mesh components with cores eventually merge, all cores that are active in the mesh send periodical messages to each other, forcing routers along the path that are not members to join the mesh. These messages are *core explicit joins* (CEJ) that specify the multicast group, the intended relay of the CEJ, the intended core, and a gap flag. The flag is an information used by the receiver of a CEJ to determine whether there are non-members in the path between two cores. When the flag is kept reset all along the path between the two cores, no acknowledgment to CEJ needs to be sent back.

A router receiving the CEJ with the gap flag set to 0 forwards the CEJ to the next relay if (a) it is the specified relay and (b) it has a path to the specified core. Furthermore, if the relaying router is not a member of the mesh, it sets the gap flag to 1 in its CEJ.

A router receiving the CEJ with the gap flag set to 1 sends an ACK if: (a) it is the intended relay, (b) it is already a member of the specified group, and (c) it has a path to the specified core. The ACK to a CEJ is forwarded all the way back to the core that originated the CEJ; ACKs force relaying routers to join the mesh as in a PJ or a regular join. Alternatively, a router receiving the CEJ with the gap flag set to 0 forwards the CEJ to the next relay if: (a) it is the specified relay, (b) it has a path to the specified core, and (c) it is not a member of the group.

A similar mechanism is used to ensure that a connected component of a group mesh with no cores in it can merge itself with at least one other connected component with one or more cores in it. When a router that ha group members or is an anchor for other routers detects that none of its successors in its shortest path to any core of the group is part of the mesh, the router simply sends a PJ towards the nearest core.

Routers use flooding (ERSs) to reconnect the mesh only if all cores are unreachable.

VIII. Packet Forwarding over a Multicast Mesh

The basic packet forwarding scheme in CAMP consists of trying to forward multicast data packets along the paths within the mesh that first reach the member routers from the sources. The main control information in a multicast packet is:

- The address of the intended multicast group.
- The address of the sending router.
- A sequence number that is used for control functions.
- A time to live used to limit the time each packet is allowed to remain in the network. packet.

A router attached to the source host of a packet simply transmits the packet to its neighbors.

A router receiving a multicast packet without errors from a neighbor router accepts the packet only if:

- The router is a member of the multicast group specified in the packet, which is determined from the router's AT.
- If the router is a duplex router, the packet's sequence number is not in the packet-forwarding cache.
- if a simplex router, the packet's sequence number is not in the packet-forwarding cache and the neighbor sending the packet is also a simplex router

When a router accepts a packet, it adds its sequence number and the identifier of the source to its packet forwarding cache. This step prevents the same packet from being accepted more than once by the router, provided that the entries in the cache persist longer than the time it takes for packets to revisit a router. In an ad-hoc network built using commercial radios operating in an ISM band with a data rate of 1 Mbps, our experiments indicate that small packet-forwarding caches suffice (e.g., listing fewer than 40 entries), because each router receives few multicast packets per second (due to limits imposed by channel access and pacing of transmissions over multiple hops) and a successful packet takes much less than a couple of seconds to traverse the longest network path.

A router that accepts a multicast packet need not forward the packet any further. A router forwards an accepted multicast packet only if the router is an anchor in the multicast group for at least one neighbor. Note that routers with source-only hosts attached do not receive multicast traffic from other sources in the group, unless they have connectivity with duplex members.

Whether a router forwards a packet or not, the router updates its MRT with the fact that the sending router belongs to the multicast group addressed by the packet. This can allow the router to join the group through a simple announcement if it is required in the future.

It is important to note a few aspects of CAMP's packet forwarding discipline. CAMP tends to forward packets along the fastest routes from sources to receivers that can be obtained *within* a multicast mesh at the time the packet is being forwarded. If link asymmetries are not very large, the shortest paths within a mesh tend to be the same as the true shortest paths, because a mesh is built ensuring that all reverse shortest paths are part of the mesh. A rare case in which packet forwarding would not take place along the shortest paths of the mesh would occur when a given router is not an anchor for any neighbor and yet is part of the shortest path within the mesh from some source to some receiver(s).

IX. Performance Comparison

A. *Protocol Used for Comparison*

In large ad-hoc networks, no multicast protocol proposed to date that is based on sender-initiated joining is scalable with the number of nodes in the network or the number of sources and groups in the network. Examples of this type of protocols based on routing trees are DVMRP and PIM-DM; an example of this type of protocols based on graphs other than trees is FGMP [2]. The reason that these protocols are not scalable is that sources must flood either data packets or control packets to *all* the network in order to establish a routing structure. If the network size is large, or the number of groups and sources per group is large, this approach is not applicable.

Accordingly, we focus our comparison of CAMP with other protocols based on receiver-oriented joining, in which the receivers ask to join the routing structure of a multicast group. Here again, flooding of joining requests to an entire network is not scalable, and we must focus on those protocols that avoid flooding as much as possible. To date, CAMP is the only multicast routing protocol not based on trees that avoids flooding of data or control packets to establish the routing structure for a group. Therefore, for comparative purposes, we implemented a simple tree-based protocol that can be used to capture all the features of the main tree-based multicast protocols with receiver-based joining proposed or implemented to date.

We implemented a shared-tree multicast routing protocol that is similar to CBT in that it uses a single core and uses that tree to forward packets. A router in this protocol, which we denote by WTP (wireless tree-based protocol), forwards data packets only when they come from one of the children or parent of the router in the tree rooted at the core. The tree maintenance part of WTP extends the conventional shared-tree protocols like CBT and PIM-SM. In WTP, a router re-establishes its connection to the tree by looking for a new parent as soon as it detects that its previous parent has moved away.

To compare CAMP against multicast protocols based on source trees (e.g., PIM-DM), without taking into account the flooding needed to set the tree, we placed the source at the core of the shared tree in some experiments.

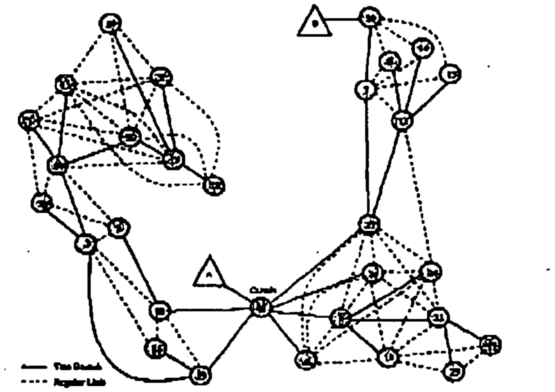

Fig. 3. Network topology used in experiments: Lines connecting nodes indicate the initial tree obtained with CBT, all routers are receivers of the multicast group, Router 16 is the core, and A and B are sources.

B. *Experiments*

CAMP rebuilds meshes at least as fast as CBT and PIM can rebuild trees and requires storage overhead similar to any protocol based on shared trees. CAMP is always loopless, forwards packets around failed links of a mesh, and is resilient to any core failure and network partitions. In contrast, CBT and PIM incur temporary loops when the unicast routing tables are inconsistent, stop packet forwarding to segments of the group after link failures until the multicast tree is rebuilt, and are vulnerable to core or rendezvous point failures. Furthermore, in a static topology, CAMP delivers packets along the shortest paths defined within a multicast mesh, which is built based on reverse shortest paths, and PIM dense mode and DVMRP build source trees based on reverse shortest paths as well. Therefore, the paths obtained in CAMP in static topologies are similar to those obtained with source trees and can be much shorter than the paths obtained with shared trees (e.g., CBT).

Accordingly, the interesting aspects for a performance comparison between CAMP and tree-based multicast protocols are the average delays and packet losses incurred due to node mobility. We ran a number of experiments to study this aspect of CAMP's performance and to compare it against the tree-based multicast approach. Figure 3 shows the topology of the dynamic network used in the simulations. The network has 30 routers, numbered from 1 to 30, and two senders, *A* and *B*. Although not indicated in the Figure, this network has high connectivity; routers have an average of six neighbors each. The links shown in the diagram illustrate the initial shared tree computed dynamically in the simulation. All nodes in the two multicast routing protocols are receivers. In CAMP, this means all nodes are *duplex* members. *Router 16* was chosen as core for all simulations.

Experiments ran for 350 seconds and the same conditions were applied to the simulation runs for CAMP and WTP; specifically, the same number of packets was sent from the given source, the same router mobility was applied, and the same MAC and routing protocols were used. The simulations used a single broadcast channel, so that the transmission of a node is received by all its neighbors. The floor acquisition multiple access (FAMA) protocol[9] was used to access the broadcast channel, and the wireless internet routing protocol (WIRP) [10] with *hop count* as distance metric was used to generate the unicast routing-table entries at routers. Radio links are bidirectional. The experiments were meant to show how much more packet loss is incurred in a tree-based scheme than in CAMP.

Two major types of experiments were run regarding mobility: one with 15 routers and the other with only 5 routers moving through the network. The speed at which mobile nodes moved randomly in all sim-

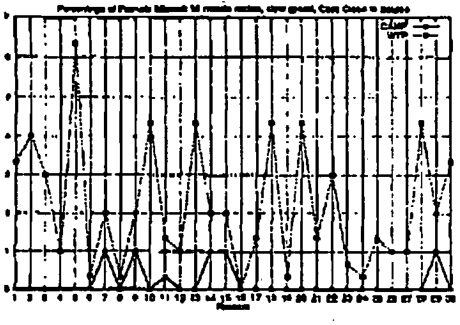

Fig. 4. Percentage of data packets missed by each router when 15 routers move at walking speed and the source of traffic is static and directly connected to the core. Mobile routers move at 0.6 meters/sec.

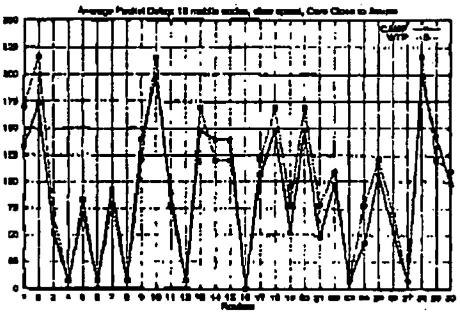

Fig. 5. Average delays of data packets received at each router when 15 routers are mobile and the source of traffic is static and directly connected to the core. Mobile routers move at 0.6 meters/sec.

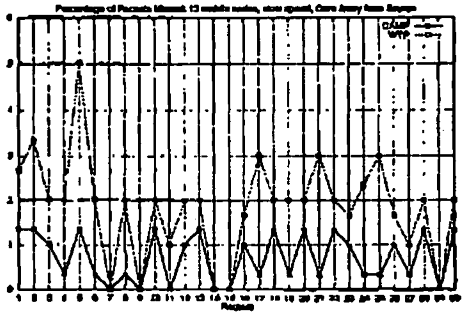

Fig. 6. Percentage of data packets missed by each router when 15 routers are mobile and the source of traffic is static and away from the core. Mobile routers move at 0.6 meters/sec.

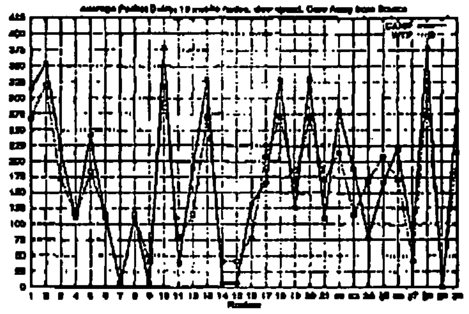

Fig. 7. Average packet delays of data packets received at each router when 15 routers are mobile and the source of traffic is static and is away from the core. Mobile routers move at 0.6 meters/sec.

ulations was 67.5 miles/hr (30 meters/sec), with the exception of one simulation experiment in which 15 routers were mobile at a walking speed of 0.135 miles/hr (0.6 meters/sec).

In most experiments, data traffic is originated either by source *A*, which is directly attached to the core (*router 16*), or by source *B*, which is attached to *router 29*. In one experiment, five routers were mobile, and the source *B* was positioned away from the core and was attached to the moving *router 17*.

The results for the case in which the source is directly attached to the core are a good basis for comparison between CAMP and multicast routing protocols based on source trees, such as PIM and DVMRP, because all routers are group members, which eliminates unnecessary overhead for such protocols.

The percentage of packets lost at a receiver is simply the amount of packets sent by the traffic source that was not seen by the specific receiver. Therefore, the smaller the percentage is, the better the protocol behaves. Obviously, the average packet delay measured at each receiver excludes lost packets.

Figure 4 to Figure 7 show the behavior of CAMP and WTP when half of the routers move at random and their speed is only 0.6 meters/sec. Because routers move slowly, the topology of the network does not change very drastically and both protocols experience few packet losses and small average delays. However, CAMP always provides fewer packet losses than the tree-based approach. In Figure 7, the average delays in WTP appear smaller in CAMP but actually they are not. This is because delays are measured based on delivered packets, and WTP delivers much fewer packets than CAMP, which biases the delay measure.

Figure 8 to Figure 11 show the behavior of both CAMP and WTP for a more dynamic environment in which five nodes move randomly at high speed (67.5 miles/hr) and the source does not move. The source in all these cases is static and is either directly attached to or distant from the core. When the source of traffic is directly connected to the core, we have the best scenario for WTP, because the shared tree becomes also a source-based tree. As the figures show, much fewer packet losses occur in CAMP than in the tree-based approach; CAMP delivers at least the same number of packets to each receiver. The average delays are comparable between CAMP and WTP for the case in which the tree is a source tree, and tend to be better in CAMP when the source is away from the core, even though the sample of delivered packets for CAMP is larger than the sample for WTP. This is due to the larger number of changes to the tree built by WTP when nodes move at higher speeds.

Figures 12 and 13 show the performance of CAMP and WTP when the source is one of the nodes moving at high speed. The negative impact of the source's mobility in a tree-based approach is evident; while in CAMP routers miss less than 5% of the packets, in WTP routers miss at least 12% and six routers miss more than 35% of the packets. This is a direct consequence of the fact that a multicast tree breaks easily with node mobility and the source of packets is the one forcing the tree to change.

Figures 14 and 15 show the comparison between CAMP and WTP when half of the 30 routers are mobile and the source of data traffic is static and directly connected to the core of the multicast group. The percentage of packet losses in CAMP is less than 15% at any receiver and is never higher than the loss rate in the tree-based approach; in contrast, loss rates in WTP are at least 60% in 10 receivers and higher than 40% in more than half half the routers, with only six routers receiving the same number of packets as in CAMP. Figure 15 illustrates the delay experienced by those packets that reach a receiver; a one-to-one com-

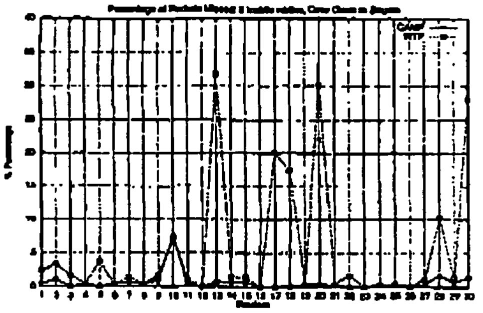

Fig. 8. Percentage of multicast data packets missed by each router when only five routers move at high speed and the source of traffic is static and directly connected to the core.

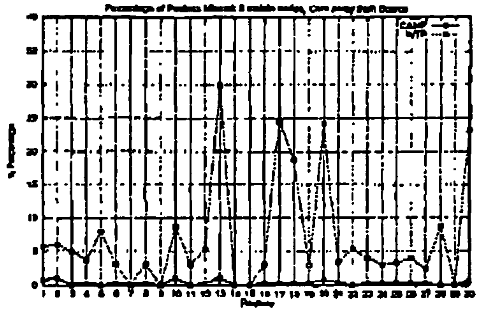

Fig. 10. Percentage of multicast data packets missed by each router when only five routers move at high speed and the source of traffic is static and is away from the core.

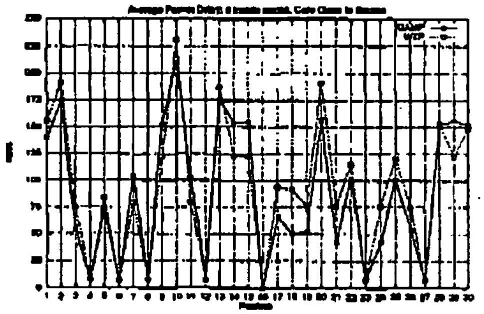

Fig. 9. Average delay of multicast data packets received at each router when only five routers move at high speed and the source of traffic is static and directly connected to the core.

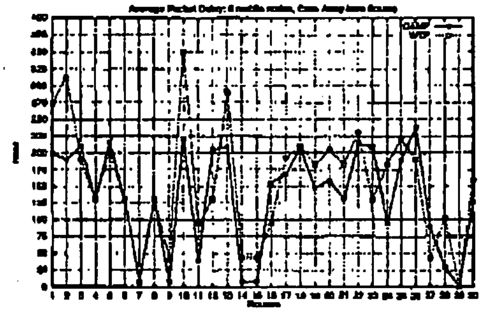

Fig. 11. Average delay of multicast data packets received at each router when only five routers move at high speed and the source of traffic is static and is away from the core of the multicast group.

parison cannot be made accurately, because CAMP delivers many more packets to the majority of the receivers. However, for those receivers that experience the same packet loss in CAMP and WTP (routers 4, 5, 8, 12, 21, and 24) the average delays in both protocols are almost the same.

Recall that in WTP and in *any* tree-based protocol for ad-hoc networks, routers must attempt to reconnect the tree as soon as possible every time a router loses its parent in the shared tree. Every time the unicast routing protocol warns WTP about a neighbor being removed from the unicast routing table, the protocol sends a join request to the new successor to the core, trying to re-establish its connection to the tree. Figure 16 shows a much more pronounced performance difference between CAMP and WTP. Packet losses in CAMP are similar to the case in which the source and the core are colocated, which is to be expected because the location of cores is irrelevant for packet forwarding in CAMP. In contrast, all receivers experience much more packet losses with the generic shared-tree protocol than they do with CAMP, and 24 receivers have packet-loss rates higher than 40%. The reason for that poor behavior of WTP is the strong dependency it has on consistent unicast routing tables to provide a loop-free shared tree. The unicast routing protocol used in the experiments may create temporary loops shortly after links go down. Because WTP makes decisions regarding tree reconnection shortly after links go down, the shared tree becomes vulnerable to loops, which leads to the larger packet-loss rate. This fact shows the difficulties brought up when packet forwarding is dictated by a strict delivery structure like a shared tree in a dynamically changing environment. Protocol behavior in the presence of temporary loops in unicast routing also illustrates CAMP's survivability.

The same problem observed in the generic shared-tree protocol when the network is more dynamic would occur in a mobile extension of PIM-SM, i.e., trying to re-establish link branches very frequently (to keep packets flowing) when routing tables are inconsistent.

X. Conclusions

We have introduced the first multicast routing protocol based on a routing structure other than trees that does not require flooding an entire network to set up its routing structure. CAMP consists of the maintenance of multicast meshes and loopless packet forwarding over such meshes. Within the multicast mesh of a group, packets from any source in the group are forwarded along the shortest paths defined with the mesh from the source to the receivers. CAMP guarantees that, within a finite time, every receiver of a multicast group has a reverse shortest path to each source of the multicast group, which is used to reduce the sub-optimality of the paths traversed within a mesh compared to the true shortest paths, which could include nodes that are not part of the mesh.

CAMP scales very well, because it does not require sources or receivers to flood the entire network with control or data packets as long as there are cores available. Simulation experiments show that CAMP easily outperforms tree-based multicast protocol in dynamic networks.

References

[1] A. Ballardie, P. Francis, and J. Crowcroft, "Core Based Trees (CBT): An Architecture for Scalable Inter-Domain Multicast Routing," *Proc. ACM SIGCOMM'93*, pages 85–95, October 1993.

[2] E. Bommaiah, M. Liu, A. McAuley, and R. Talpade, "AMRoute: Adhoc Multicast Routing Protocol," Internet Draft, http://www.ietf.org/internet-drafts/draft-talpade-manet-amroute-00.txt

[3] C. Chiang and M. Gerla, "On-Demand Multicast in Mobile Wireless Networks," *Proc. IEEE ICNP 98*, Austin, Texas, October 14-16, 1998.

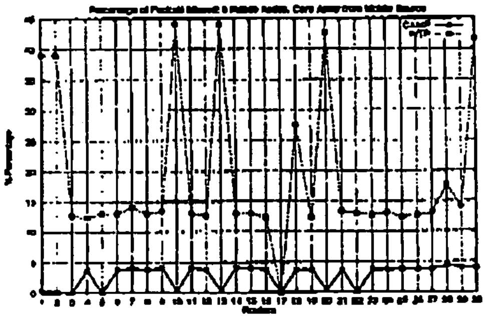

Fig. 12. Percentage of data packets missed by each router when the source (connected to router 17) and four other routers move at high speed.

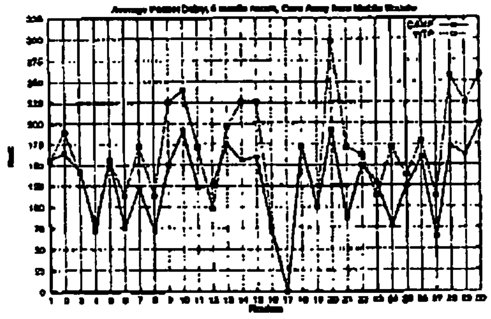

Fig. 13. Average delays of data packets received at each router when the source (connected to router 17) and four other routers move at high speed.

[4] M.S. Corson and S.G. Batsell, "A Reservation-Based Multicast (RBM) Routing Protocol for Mobile Networks: Overview of Initial Route Construction," *Proc. IEEE INFOCOM '95*, Boston, MA, April 1995.

[5] M.S. Corson and V. Park, "A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks," *Proc. IEEE INFOCOM '97*, Kobe, Japan, April 1997.

[6] S. Deering, "Host extensions for IP multicasting," RFC-1112, August 1989.

[7] S. Deering, C. Partridge and D. Waitzman, "Distance Vector Multicast Routing Protocol," RFC-1075, November, 1988.

[8] S. Deering et al. "An Architecture for Wide-Area Multicast Routing," *Proc. ACM SIGCOMM'94*, Cambridge, MA, 1994.

[9] C. L. Fullmer and J.J. Garcia-Luna-Aceves, "Solutions to Hidden Terminal Problems in Wireless Networks," *Proc. ACM SIGCOMM 97*, Cannes, France, September 14-18, 1997.

[10] J.J. Garcia-Luna-Aceves, C.L. Fullmer, E. Madruga, D. Beyer and T. Frivold, "Wireless Internet Gateways (WINGS)". *Proc. IEEE MILCOM'97*, Monterey, California, November 2-5, 1997.

[11] G.S. Lauer, "Packet Radio Routing," *Chapter 11, Routing in Communications Networks*. (M.Steenstrup, ed.), Prentice-Hall, Englewood Cliffs, NJ, 1995.

[12] K. Makki, N. Pissinou, and O. Frieder, "Efficient Solutions to Multicast Routing in Communication Networks," *Mobile Networks and Applications*, ACM/Baltzer, Vol. 1, No. 2, 1996.

[13] S. Murthy and J.J. Garcia-Luna-Aceves, "An Efficient Routing Protocol for Wireless Networks," *Mobile Networks and Applications*, ACM/Baltzer, Vol. 1, No. 2, 1996.

[14] C.E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," *Proc. ACM SIGCOMM '94*, London, UK, 1994.

[15] C. Shields and J.J. Garcia-Luna-Aceves, "The Ordered Core-Based Tree Protocol," *Proc. IEEE INFOCOM '97*, Kobe, Japan, April 1997.

[16] M. Spohn and J.J. Garcia-Luna-Aceves, "Scalable Link-State Internet Routing," *Proc. VI IEEE International Conference on Network Protocols (ICNP '98)*, Austin, TX, October, 1998.

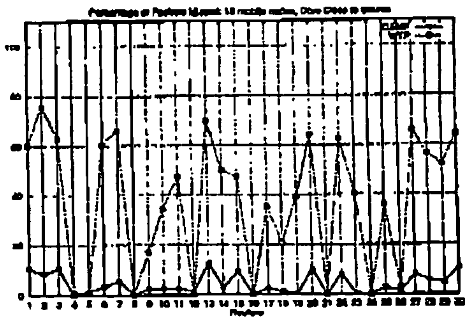

Fig. 14. Percentage of multicast data packets missed by each router when 15 of the routers move at high speed and the source of traffic is directly connected to the core.

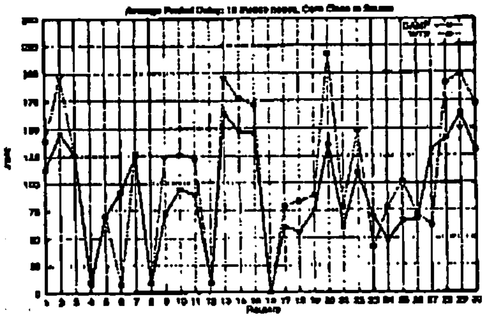

Fig. 15. Average delays of multicast data packets received at each router when 15 of the routers move at high speed and the source of traffic is directly connected to the core of the multicast group.

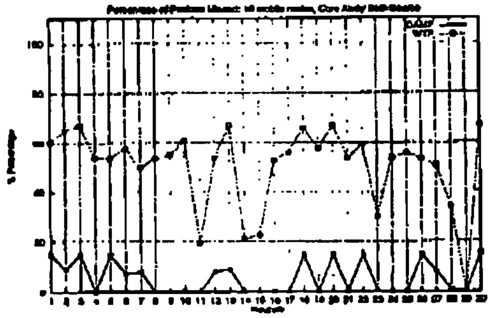

Fig. 16. Percentage of multicast data packets missed by each router when 15 of the routers move at high speed and the source of traffic is away from the core of the multicast group.